

Express Mailing Label No.: ER211533099US

PATENT APPLICATION

Docket No.: 1300.2.16

IBM Docket No.: RPS920020138US1

UNITED STATES PATENT APPLICATION

of

Timothy W. Crockett

Paul D. Kangas

and

Rex Lamont Wills

for

KEY CODE FILTER APPARATUS AND METHOD

KEY CODE FILTER APPARATUS AND METHOD

BACKGROUND OF THE INVENTION

1. The Field of the Invention

This invention relates to digital computers, and more particularly, to unique apparatus and methods for blocking selected key codes, sequences of key codes, and combinations of key codes transmitted between a keyboard and a host computer.

2. The Relevant Art

Kiosks of various types and providing a wide variety of services, are currently increasing in popularity. Most kiosks operate using standard computer hardware and include some type of input device such as a keyboard. When designing a kiosk for use by the public, it is often desired to prevent customers from entering certain keys, key sequences, or key combinations from the keyboard. For example, if the input device is a standard keyboard, it may be desired to prevent a user from entering a key combination such as a "CTRL-ALT-DEL" sequence. This key sequence may enable a user of a kiosk to access shut down or reboot commands, or may allow undesired access to applications or data.

Currently, offending keys are blocked using either hardware or software. A typical hardware method is to design a keyboard that does not offer undesired keys to a user. This limits the choices available to a designer and may add expense by requiring a customized keyboard. One typical software solution is to alter drivers or system BIOS on a kiosk host computer so that undesired key codes are ignored. This solution can be impractical since a designer may then become responsible for writing and maintaining keyboard driver code for each operating system used.

Given the drawbacks of current solutions, apparatus and methods are needed to improve currently practice methods for blocking or filtering out selected key codes.

1 Consequently, a need exists for improved apparatus and methods that block or filter out
2 selected keys, sequences of keys, or combinations of keys that a user may input from a
3 keyboard or other input device. What is further needed are such apparatus and methods that
4 are easily installed with an existing keyboard and host computer without requiring
5 modification of an operating system or installing additional applications on the host
6 computer.

7 In particular, what is needed is an apparatus, physically independent from the host
8 computer and keyboard, that may simply be inserted therebetween to monitor combinations
9 and sequences of key presses, in addition to keys pressed alone, which would provide the
10 desired key code filtration. Accordingly, it would also be an advance in the art to provide a
11 filtration device that could be easily reprogrammed or updated to add, delete, or modify the
12 set of selected keys codes to be blocked.

SUMMARY OF THE INVENTION

Accordingly, an apparatus in accordance with the present invention provides a key code filter to filter out selected key codes transmitted between an input device such as a keyboard and a host computer. It is realized that normally, between an input device and host computer, there are several layers of keyboard code translation. For example, translation exists between keyboard to controller codes, controller to system codes, and resultant ASCII codes. For the sake of the present discussion, the term "key code" will be used interchangeably for any of the aforementioned key codes to provide simplicity, although one term used in technical literature is typically "Kscan" code for codes transmitted between controllers resident in the keyboard and host computer.

In one aspect of the present invention, an apparatus for filtering out selected key codes includes an input port to receive key codes transmitted from an input device such as a keyboard. The input port is operably connected with a key code filter whose function is to block selected key codes, sequences of key codes, and combinations of key codes, as selected by a kiosk or computer station designer. Once the key code filter filters out selected key codes, remaining unblocked key codes are transmitted through an output port to the host computer.

In certain embodiments, the key code filter is physically and functionally independent from the input device and the host computer. However, in other embodiments, the key code filter is located proximate the host computer or the input device, or is actually located within the housing of either.

The key code filter in one embodiment includes a processor configured to process executable and operational data. The key code filter may also include a memory device operably connected to the processor to store the executable and operational data. In one embodiment, the executable and operational data includes a filtering module programmed to filter out desired key codes.

Most keyboards produce key codes that can be classified into two different types. A

1 first type of key code corresponds to the pressing of a particular key on the keyboard. For the
2 sake of the present discussion, this type of key code will be referred to as “make key code.”
3 A second type of key code is transmitted from the keyboard to the host computer when the
4 same key is released. This type of key code will be referred to as a “release key code” in this
5 specification.

6 As stated herein, in certain embodiments, the filtering module associated with the key
7 code filter comprises executable code and operational data stored in memory associated with
8 a processor, such as may be found in a microcontroller. The filtering module may include a
9 rules table containing a list of key codes and associated rules for each key code. Each rule is
10 used to determine whether a key code is blocked or transmitted. Some key codes may be
11 blocked under all circumstances. Other key codes may only be blocked when they are part of
12 a particular sequence of key codes. Yet other key codes are blocked only when they are part
13 of a selected combination of key codes.

14 In certain embodiments, the filtering module also includes a “make list.” The make
15 list is a current tally of make key codes (corresponding to pressed keys) that have been
16 received by the key code filter and transmitted on to the host computer, for which
17 corresponding release key codes have yet been received by the key code filter. That is, if the
18 key code filter receives a “make code” from the input device, the make code is forwarded to
19 the host computer if allowed by the corresponding rule.

20 An entry corresponding to the make code is entered in the make list until a
21 corresponding “release code” is received by the key code filter. When the release code is
22 received, the release code is forwarded to the host computer and the entry is removed from
23 the make list.

24 A function of the make list is to keep track of make codes that are received by the key
25 code filter so that combinations of keys can be detected. That is, if a user presses a first key
26 on the keyboard and proceeds to press a second key before releasing the first, this
27 combination can be detected by examining make codes already in the make list.

1 Likewise, in certain embodiments, the filtering module includes a block list
2 containing a list of make key codes that have been blocked, for which corresponding release
3 key codes have not yet been received by the key code filter. That is, if a "make code" has
4 been received and has been blocked from transmission to the host computer, the make code
5 is stored in a "block list" until a corresponding "release code" is received by the key code
6 filter. Since it may confuse a host computer if a "make code" is blocked, and the
7 corresponding "release code" is received by the host computer, the block list functions to
8 ensure that a release code, corresponding to a blocked make code, is blocked.

9 In certain embodiments, the functionality of key code filter is provided by a
10 microcontroller. One advantage of using a microcontroller is that most of the functionality of
11 the key code filter is provided by software loaded thereon. A microcontroller includes many
12 needed components in an integrated package, such as a processor, memory, I/O ports, an
13 oscillator, and the like, thereby simplifying the design of the key code filter.

14 Since microcontrollers and other circuitry within a keyboard operate using power
15 provided by the host computer, the key code filter may also operate using power from the
16 host computer. Since the key code filter is inserted between the host computer and the input
17 device, the key code filter simply taps into the power source provided to the keyboard.

18 In certain embodiments, the functionality of the key code filter is exclusively
19 provided by hardware. That is, the circuitry within the key code filter is a dedicated hard-
20 wired solution that requires no software. Implementations of this type may be useful in cases
21 where a standardized set of key codes is blocked, and the device will not likely need
22 reprogramming. Moreover, it is possible that a hardware implementation could be less
23 expensive in certain cases, since certain hardware components and software development are
24 not needed.

25 In other embodiments, the functionality of the key code filter is provided by a
26 combination of hardware and software. That is, a microcontroller or other programmable
27 device is loaded with software to perform key code. In this embodiment, software may be

1 upgraded, as needed, or other software may be loaded or reprogrammed to add or delete
2 specific key codes from those on the filtered list.

3 In another aspect of the present invention, a method for filtering out selected key
4 codes transmitted between an input device and a host computer intercepts key codes
5 transmitted between an input device and a host computer. Once the key codes are
6 intercepted, the method includes blocking, selected key codes, sequences of key codes, and
7 combinations of key codes. The method forwards unblocked key codes to the host computer.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features of the present invention will become more fully apparent from the following description, taken in conjunction with the accompanying drawings. Understanding that these drawings depict only typical embodiments in accordance with the invention and are, therefore, not to be considered limiting of its scope, the invention will be described with additional specificity and detail through use of the accompanying drawings in which:

Figure 1A is a schematic block diagram illustrating one embodiment of a key code filter inserted between a keyboard and a host computer in accordance with the invention;

Figure 1B is a schematic block diagram illustrating one embodiment of the key code filter of Figure 1A;

Figure 2 is a schematic block diagram illustrating one embodiment of various components providing functionality within the keyboard, the key code filter, and the host computer;

Figure 3 is a schematic block diagram illustrating one embodiment of various functional modules contained within the memory of the key code filter;

Figure 4 is a schematic block diagram illustrating one embodiment of an algorithm for filtering key codes;

Figure 5 is a chart illustrating an example of a rules list for filtering out specific key codes, sequences of key codes, and/or combinations of key codes in accordance with the invention;

Figure 6 shows two charts illustrating two examples of key sequences that may be received and processed by the key code filter;

Figure 7 is a schematic diagram illustrating one embodiment of circuitry that is used with the key code filter to facilitate the reception and transmission of data and clock signals between the keyboard and the host computer; and

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

Figure 8 is a timing diagram illustrating one embodiment of timing and relationships of clock and data signals transmitted between a keyboard and a host computer.

DETAILED DESCRIPTION OF THE INVENTION

It will be readily understood that the components of the present invention, as generally described and illustrated in the Figures herein, could be arranged and designed in a wide variety of different configurations. Thus, the following more detailed description of the embodiments of systems and methods in accordance with the present invention, as represented in Figures 1 through 8, is not intended to limit the scope of the invention, as claimed, but is merely representative of certain examples of presently contemplated embodiments in accordance with the invention. The presently described embodiments will be best understood by reference to the drawings, wherein like parts are designated by like numerals throughout.

Many of the functional units described in this specification have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, modules may be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module. For example, a module of executable code could be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices.

Modules may also be implemented in hardware as electronic circuits comprising custom VLSI circuitry, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

1 Similarly, operational data may be identified and illustrated herein within modules,
2 and may be embodied in any suitable form and organized within any suitable type of data
3 structure. The operational data may be collected as a single data set, or may be distributed
4 over different locations including over different storage devices, and may exist, at least
5 partially, merely as electronic signals on a system or network.

6 Referring to Figures 1A and 1B, in one embodiment, a key code filter 102 is inserted
7 between a keyboard 100 and a host computer 104. While the particular embodiment of a
8 keyboard is given here by way of example, it should be readily recognized that the present
9 invention may be used with other types of input devices, including voice recognition
10 modules, touch pads, light activation devices, and the like. Similarly, the host computer may
11 be any computing device, including but not limited to an Internet or networking terminal, a
12 main frame computer, distributive computing, a personal computer, a hand-held computer,
13 and a processor in a customer services kiosk.

14 A communications link 106, normally coupled directly to the host computer 104 is
15 coupled to the key code filter 102. Likewise an output line 108 couples the key code filter
16 102 to the host computer 104. As illustrated, the key code filter 102 is an independent device
17 that is physically separate from the keyboard 100 and the host computer 104. However, in
18 other contemplated embodiments, the key code filter 102 is located proximate either the
19 keyboard 100 or the host computer 104, or is actually contained within the housing of either
20 device 100, 102. The key code filter 102 may, in certain embodiments be integrated with the
21 circuitry of either the keyboard 100 or the computer 104.

22 In one embodiment depicted in Figure 1B, the key code filter 102 includes an I/O port
23 112 for communicating with the keyboard 100 and an I/O port 114 for communicating with
24 the host computer 104. As depicted, the I/O ports 112, 114 are operably connected to a
25 filtering module 110. The filtering module 110 is configured to receive key codes from the
26 I/O ports 114, recognize the key codes, compare those key codes to a list of restricted key
27

1 codes, sequences of key codes, or combinations of key codes and restrict those key codes
2 from transmission to the host computer 104.

3 Referring to Figure 2, a key code filtering system 200 depicts specific embodiments
4 of the elements of Figures 1A and 1B in greater detail. For example, the depicted key code
5 filter 102 includes one or more input and output ports 112, 114 that receive and transmit data
6 and clock signals 106, 108 to and from the keyboard 100 and the host computer 104. The
7 input and output ports 112, 114 transmit the data and clock signals to a processor 210 or
8 CPU 210. The CPU 210 processes the incoming key codes, and either blocks the key codes,
9 or forwards them to the host computer 104.

10 As depicted in the embodiment of Figure 2, the keyboard 100 contains an internal
11 controller 201. The controller 201 is effective to sense when keys are pressed and released
12 on the keyboard 100. When a key is pressed, the controller 201 outputs a “make” key code
13 corresponding to the pressed key. Likewise, when the same key is released, the keyboard
14 controller 201 outputs a corresponding “release” key code.

15 Both the make and release key codes are represented by a certain number of bits or
16 bytes. In order to transmit the bits or bytes from the keyboard, a clock line 204 may be used
17 to coordinate the transfer. In one embodiment, the clock line 204 indicates the first bit of a
18 key code, times to sample intervening bits, a stop bit, as well bits used for error checking,
19 carried on a data line 202. An I/O port 112 may receive the incoming key codes from the
20 controller 201.

21 As depicted, the key codes are transmitted to a CPU 210 on a system bus 212. The
22 CPU 210 processes the key codes and determines whether they are to be blocked or
23 forwarded to the host computer 104. If the key code filter 102 is a programmable device, the
24 key code filter 102 includes a memory module containing executable and operational data
25 processable by the processor 210. The executable and operational data is effective to filter
26 out selected key codes, sequences of key codes, or combinations of key codes, as desired for
27 a particular kiosk system or general computing device.

1 The memory module 216 may include a volatile portion (RAM) 218 and a nonvolatile
2 portion (ROM) 220. When the key code filter 102 is powered up, selected executable and
3 operational data may be transferred from the ROM 220 to the RAM 218. In addition, in
4 certain embodiments, the ROM 220 may be reprogrammed as needed to add or modify key
5 codes to be blocked, or to update the executable and operational data.

6 The key code filter 104 may include other components as necessary, including but not
7 limited to a timer 222, an oscillator 224, or any other components 226 that may or may not be
8 needed. Certain microcontrollers may provide any or all of components used to implement
9 the key code filter 102, in addition to certain components that may be present but not used.

10 An I/O port 114 receives the unblocked key codes from the CPU 210 on a bus 214,
11 and forwards those key codes on to the host computer 104 or workstation 104 by way of data
12 lines 228 and clock lines 230. The host computer 104 includes a dedicated controller 232
13 located on the motherboard to interface with a keyboard 100, or in this case, the key code
14 filter 102. The keyboard controller 232 acts as an interface between the keyboard 100 and a
15 CPU 234, located within the host computer 104.

16 For example, when the keyboard controller 232 receives a key code from the key code
17 filter 102, the keyboard controller 232 may send an interrupt to an interrupt controller 252.
18 The interrupt controller 252 may then interrupt those tasks currently being executed by the
19 CPU 234. The CPU 234 may then respond to the interrupt by retrieving the key code and
20 transmitting it to its destination. Because current CPUs 234 operate at such great speeds,
21 even relatively numerous and frequent interruptions caused by a skilled typist will cause very
22 little performance degradation of the CPU 234.

23 The host computer 104 includes volatile and nonvolatile memory that may be used as
24 data buffers 242, memory buffers 242 used to temporarily hold key codes while being
25 transferred to and from the CPU 234, and interrupt handlers 244 used to respond to incoming
26 key codes. The memory 238 may also include the operating system 246 to act as an interface
27

1 between applications 248 attempting to access keyboard information. A system control
2 module 233 may be used to send control commands to the motherboard controller 232.

3 The depicted system 200 of Figure 2 is merely provided as a high level aid for
4 understanding the function of the key code filter 102 with respect to the keyboard 100 and
5 the host computer 104, and is in no way intended to limit the scope of the present invention.
6 In fact, there are many different ways the key code filter 102 could be designed, while still
7 providing a filter 102 that can operate independently from the keyboard 100 and the host
8 computer 104.

9 With respect to Figure 3, the memory module 216, as illustrated in Figure 2, includes
10 various executable code and operational data to provide certain functions of the key code
11 filter 102. For example, in certain embodiments, the memory module 216 includes a filter
12 module 300. The filter module 300 includes the necessary logic to block selected key codes,
13 while transmitting other selected key codes.

14 In certain embodiments, the filter module 300 includes a rules table 302. The rules
15 table 302 includes a list of rules corresponding to each key code that can be received by the
16 key code filter 102. The rules for certain key codes may be to transmit the key code. The
17 rules for other key codes may be to block those key codes. Yet other rules may block certain
18 key codes only if other key codes have been received first, indicating that a certain
19 combination of key codes is to be blocked.

20 In selected embodiments, the filter module 300 includes a “make list” 304. The make
21 list 304 is a list of make key codes that have been transmitted (e.g. not blocked), for which a
22 corresponding release code has not yet been received. Likewise, in certain embodiments, the
23 filter module 300 includes a “block list” 306 that lists each make code that has been blocked,
24 and for which a corresponding release code has not yet been received. The make list 304 is
25 used to keep track of keys that have been pressed but not yet released, in order to monitor
26 combination of keys that are pressed simultaneously. For example, certain keys may be
27 transmitted when pressed or released alone. However, these same keys may be blocked if

1 pressed in combination with other keys. The make list 304 aids in keeping track of these
2 combinations.

3 Likewise, if certain make key codes are blocked, it may be confusing to a host
4 computer if a corresponding release code is received. The block list 306 is used to keep track
5 of make key codes that have been blocked. In this manner, when the corresponding release
6 key code is received, it is also blocked to avoid this type of confusion.

7 The memory module 216 may also include an I/O module 308 that may be
8 responsible for inputting and outputting key codes from the key code filter 102. This module
9 308 may help coordinate the input and output of key codes in accordance with timing
10 protocols that are used between a keyboard 100 and a host computer 104.

11 Referring to Figure 4, in one embodiment, a method 400 for filtering key codes
12 begins 402 and then proceeds to receive 404 a key code from the keyboard 100. A make test
13 406 determines whether the key code is a make key code. If the key code is a make key code,
14 a rule test 408 tests whether the rule corresponding to the make key code is true. In this
15 example, if a rule is "true," then a key code is blocked and is not transmitted to the host
16 computer 104. Thus, if at the rule test 408, the rule is true, the release code corresponding to
17 the make code is added 412 to the blocked list and the process is ended 422 without
18 transmitting the make code to the host computer 104.

19 If, however, at the rule test 408 the rule is false, the make key code is added 414 to
20 the make list and the make key code is sent to the keyboard host in the host computer 104.

21 Returning back to the make test 406, if the key code received at step 404 is not a
22 make key code, then it is a release key code, and the process 400 is passed to a block test
23 410. If at the block test 410, the release key code is in the block list, the release key code is
24 removed 418 from the blocked list and the process 400 is ended 422 without transmitting the
25 release code to the host computer 104. If, however, at the block test 410, the release code is
26 not in the blocked list, the corresponding key code is removed 416 from the make list and the
27 release code is transmitted 420 to the keyboard host in the host computer 104. This process

1 400 may be more easily understood from examples presented below as part of the discussion
2 of Figure 6.

3 Referring to Figure 5, in one embodiment, a rules table 302 includes a key column
4 502, a key code column 504, and a rule column 506. For example a "Del" key 508 (Delete)
5 may have a key code 0X17. A rule corresponding to the "Del" key may be "Alt and Cntr."
6 Thus, if the "Del" key is pressed while the "Alt" and "Cntr" keys are also pressed, then the
7 rule will be true, and the "Del" key will be blocked. This prevents a user from pressing the
8 "CTRL-ALT-DEL" sequence, which in many systems would shut down or reboot the host
9 computer 104. However, by the same token, if the "Del" key is pressed and one or both of
10 the "Alt" and "Ctrl" keys are not pressed, then the rule will be false and the key will be
11 passed on and not blocked.

12 In a similar manner, if a user presses the "0" key 510, the rule for this key is
13 "FALSE," so the key code is transmitted to the host computer 104. If a user presses the "c"
14 key 512, the key is only blocked if it is pressed while both keys "Alt" and "Ctrl" are pressed,
15 or if the "Ctrl" key is pressed by itself. If a user presses the "F1" key 514, the corresponding
16 rule is true, and the key code is blocked from transmission. Likewise, if a user presses any of
17 the "R-Alt" (right hand Alt key) 516, "R-Ctrl" 518, "L-Alt" 520, or "L-Ctrl" 522 keys, the
18 rules for these keys 516, 518, 520, 522 are false, and the codes are transmitted to the host
19 computer 104. By reprogramming the key code filter 102, rules may be added, deleted, or
20 modified from the rules table 302 as needed, to block or transmit selected key codes.

21 Referring to Figure 6, while referring generally to Figure 4, a "CTRL-ALT-DEL"
22 key sequence 600 may be used by way of example to illustrate the function of the present
23 invention. The key sequence 600 may be characterized by an event column 602, a make list
24 column 604, a block list column 606, and a column 608 listing the keys sent to the host
25 computer 104. For example, the key sequence 600 may begin with the pressing 610 of the
26 "Ctrl" key. In this case, the rule for the "Ctrl" make key code is "FALSE," so the "Ctrl"
27

1 make key code is transmitted to the host computer 104, and an entry corresponding to the
2 "Ctrl" key is placed in the make list.

3 A second event 612 may be the pressing of the "Alt" key 612. The rule for the "Alt"
4 key is also "FALSE," so the Alt make key code is transmitted to the host computer 104 and
5 an entry corresponding to the "Alt" key is added to the make list.

6 However, when the "Del" key is pressed 614, since the "Ctrl" and "Alt" keys are also
7 pressed, the rule for this key is "TRUE," so this key is blocked and an entry is made in the
8 block list. When the "Del" key is released, a "Del" release key code is received. Since an
9 entry corresponding to the "Del" key is in the blocked list, the "Del" release key code is
10 blocked and the "Del" entry is removed from the block list 606.

11 The release of the "Alt" key is the next event 618 to occur. When this occurs, the
12 "Alt" release key code is transmitted to the host computer 104, and the "Alt" entry is
13 removed from the make list. Likewise, the release of the "Ctrl" key is the next event 620 to
14 occur. When this occurs, the "Ctrl" release key code is transmitted to the host computer 104
15 and the "Ctrl" entry is removed from the make list. Thus, in this manner, the "CTRL-ALT-
16 DEL" key sequence is blocked from the host computer 104.

17 In a second key sequence 650, shown by way of example, the "F1" key is pressed
18 652. Since the rule for this key is "TRUE," the "F1" make key code is blocked from
19 transmission, and an "F1" entry is made in the block list 606. When the "F1" key is released
20 654, the "F1" release key code is blocked from transmission and the "F1" entry is removed
21 from the "block" list.

22 Referring to Figure 7, the key code filter 102 may be implemented using various
23 circuit components in order to effectively receive and transmit information to and from the
24 keyboard 100 and the host computer 104. For example, the clock and data lines 202, 204,
25 228, 230 are used bi-directionally in Figure 7 to enable communication between the keyboard
26 100 and the host computer. In order to prevent contention caused by either the keyboard 100
27

1 or the host computer 104 trying to communicate at the same time as the key code filter 102,
2 buffers 700a-h may be used.

3 A single set of communication lines may be used to share signaling in two directions
4 by using open collector buffers 700a-h and software control to monitor activity on the bi-
5 directional lines 202, 204, 228, 230. The buffers 700a-h provide isolation between
6 respective transmitting and receiving lines on the keyboard microcontroller 201 and the
7 computer keyboard controller 232. Each of the buffers 700a-h may be an open collector
8 buffer 700a-h. The properties of the open collector buffers 700a-h are such that the buffers
9 700a-h never output a "high" voltage. Rather, the buffers 700a-h use two states: ground
10 potential or high impedance (open circuit). The resistors 702a-h serve to prevent shorts
11 between system voltage and ground potential.

12 Referring to Figure 8, typically, the keyboard controller 232, located on the
13 motherboard, communicates with the keyboard 100 over a serial line 106, 108. Likewise, the
14 key code filter 102 communicates with both the keyboard 100 and the host computer 104 in
15 the same manner. A clock line 230 carrying a synchronized clock signal 802a is provided
16 from the motherboard controller 232 to the key code filter 102. Likewise, a data line 228
17 carrying a serial data frame 804a of 11 bits may be transmitted consisting of a start bit 805,
18 eight data bits 806, an odd parity bit 808, and a stop bit 810.

19 In an idle state, both the data 804a and clock 802a signals are high. To begin sending
20 data to the motherboard keyboard controller 232, the key code filter 102 sends a start bit 805
21 on the data line 228. The motherboard keyboard controller 232 responds by starting the
22 clock signal 802a high, with a first clock pulse 812a going low. The clock 802a continues
23 and the key code filter 102 samples each data bit 806a, 806b as the clock 802a goes low. At
24 the 11th clock cycle, the key code filter 102 sends a stop bit 810 and the clock line 802a
25 returns to an idle high state.

26 The process described may also be used by the host computer 104 to send data to the
27 key code filter 102. For example, in an idle state, both the data 804b and clock 802b signals

1 are high. To begin sending data to the key code filter 102, the motherboard keyboard
2 controller 232 sends a start bit 812b on the data line 228. The key code filter 102 responds
3 by starting the clock signal 802b, with the first pulse 812b going low. Under normal
4 operation, the clock 802b is continued with each data bit 814a, 814b, 816 being sampled as
5 the clock 802b goes low. At the 11th clock cycle, the motherboard keyboard controller 232
6 sends the stop bit 818 and the clock signal 802b returns to its high idle state.

7 Communication between the key code filter 102 and the keyboard 100 functions in
8 much the same way as between the key code filter 102 and the keyboard motherboard
9 controller 232, as has been described. In selected embodiments, the transmission of key
10 codes from the computer 104 to the keyboard 100 is transparent with no filtering. In other
11 embodiments, the transmission of key codes therebetween is filtered in the same manner as
12 from the keyboard 100 to the computer 104. The described method of communication
13 between components 100, 102, 104 is simply one example of how communication may
14 occur. Obviously, as newer systems are developed, the method of communication may
15 change. However, any suitable method of communication between the keyboard 100, the key
16 code filter 102, and the host computer 104 are certainly considered to be within the scope of
17 the present invention.

18 The present invention may be embodied in other specific forms without departing
19 from its essence or essential characteristics. The described embodiments are to be
20 considered in all respects only as illustrative, and not restrictive. The scope of the invention
21 is, therefore, indicated by the appended claims, rather than by the foregoing description. All
22 changes within the meaning and range of equivalency of the claims are to be embraced
23 within their scope.

24 What is claimed is:
25
26
27